Tuijin Jishu/Journal of Propulsion Technology
ISSN: 1001-4055
Vol. 45 No. 2 (2024)

Digital Forensics: Deep Learning's Approach to Deepfake Image Identification

Buvaneshwaran P[1,3], V Ramesh Babu[2,3], S Geetha[2,3]
[1]Postgraduate Student, [2]Professor,
[3]Department of Computer Science and Engineering, Dr. M.G.R. Educational and Research Institute, Chennai.

Abstract -The paper propose a deepfake detection method leveraging EfficientNet, InceptionResNet, and MobileNet architectures. EfficientNet offers computational efficiency, while InceptionResNet provides depth and feature richness, and MobileNet balances accuracy and computational resources. The models are trained on diverse datasets using transfer learning and fine-tuning for deepfake detection. Enhanced discriminative power is achieved through attention mechanisms and feature fusion, alongside exploration of ensemble methods for improved accuracy. Evaluation on benchmark datasets demonstrates superior effectiveness compared to existing methods. The approach addresses evolving challenges in deepfake detection, showcasing versatility and adaptability for real-time applications.
Keywords: *Deepfake, Deep learning, EfficientNet, InceptionResNet, MobileNet*

1.Introduction

The advent of deepfake technology marks the beginning of a novel era fraught with challenges concerning the veracity of multimedia content and the propagation of misinformation. Deepfakes, crafted using sophisticated artificial intelligence algorithms, have transcended from obscure novelties to formidable tools capable of manipulating both visual and auditory perceptions with startling realism. To counter these challenges, innovative and adaptable solutions are imperative, particularly at the intersection of deep learning and image identification.

This research endeavors to harness the capabilities of three leading deep learning architectures—EfficientNet, InceptionResNet, and MobileNet—in the realm of deepfake identification. Each architecture offers a distinct set of attributes and computational efficiencies, providing a diverse arsenal for unraveling the intricate fabric of synthetic media. EfficientNet, renowned for its superior computational efficiency, is well-suited for real-time applications. InceptionResNet amalgamates the strengths of the Inception and ResNet architectures, yielding a deep and feature-rich model. MobileNet, tailored for mobile and edge devices, strikes a balance between accuracy and computational efficiency.

Conventional methods of identifying manipulated content often lag behind the rapid advancements in generative models, necessitating a shift towards cutting-edge technologies like deep learning. This study is driven by the urgent necessity to create solutions that are both effective and efficient in identifying deepfake images.As deepfake techniques grow increasingly sophisticated, traditional detection methods struggle to keep pace. Thus, the adoption of deep learning architectures becomes imperative to discern subtle patterns and anomalies indicative of synthetic media, all while maintaining computational efficiency—a crucial factor for real-world deployment.

The technology has the potential to move beyond its initial intentions for entertainment by influencing deeply important aspects of our lives. In a world where visual content holds sway over public opinion and societal narratives, deepfakes pose an existential threat to truth and trust. This can be seen in their capacity to provoke political unrests, undermine the credibility of journalism, and cause discord in many other sectors signifying urgent need for sophisticated techniques able to distinguish between real and manipulated.

EfficientNet, InceptionResNet and MobileNet architectures exhibits flexibility and adaptability of deep learning approaches towards addressing complex issues raised by deepfakedetection. By delving into these architectures and exploring their potential combinations, this research aims to contribute significantly to the ongoing efforts in creating robust and efficient systems for detecting manipulated images. Only through concerted efforts and interdisciplinary collaborations can we hope to mitigate the risks posed by deepfake technology and safeguard the integrity of digital media.

2. Related Works

Traditional forensic techniques have long been employed in the analysis of digital media to detect manipulations or alterations. These methods typically involve scrutinizing various aspects of the media content for inconsistencies or anomalies that could indicate tampering. Some of the common traditional forensic techniques include:

Metadata Analysis: Metadata refers to the descriptive information embedded within digital files, such as timestamps, camera settings, and location data. Forensic analysts examine metadata to identify inconsistencies or irregularities that may suggest manipulation or fabrication. For instance, a mismatch between the timestamp of a digital image and the purported time of capture could indicate tampering.

Error Level Analysis: Error level analysis involves analyzing the compression artifacts present in digital images to detect regions that may have been altered. When an image is compressed and recompressed, areas that have been modified may exhibit different error levels compared to the rest of the image. By examining these discrepancies,

forensic experts can identify potentially manipulated regions within an image.

Image Noise Analysis: Image noise refers to random variations in pixel values that are inherent in digital images. Forensic analysts utilize image noise analysis to detect inconsistencies in noise patterns across different regions of an image. Areas that have been digitally altered may exhibit irregular noise patterns that deviate from the surrounding content, indicating potential tampering.

While traditional forensic techniques have proven useful in certain scenarios, they often face limitations when dealing with sophisticated manipulation techniques, such as deepfakes generated using advanced machine learning algorithms.

In contrast, deep learning-based methods have emerged as a promising approach for detecting deepfakes because of their capacity to grasp intricate patterns and features from large datasets. These methods leverage neural networks to analyze digital media for signs of manipulation. They can be broadly categorized into two types:

Image-Based Detection: Image based deepfake detector models are learned to look for certain visual signs or artifacts, that are typical for deep fake deception. On the other hand, employing these models, the facial expression, eye or lighting and shadows for example, may be used to reveal the real from the fake pictures.

Video-Based Detection: The detection in deepfake video-focused models of such tells relies on the analysis of temporal information in videos. Such models will, thus, be able to scrutinize the movement and congruence of video parts in order to locate possible inconsistencies and determine if the specific video is indeed a deepfake.

Although deep learning-based methods have proved to be a successful tool to fighting some types of deepfakes, they have a challenge of their own. The use of such methods is mostly dependent on having huge amounts of labeled data and they are likely to have problems to identify those deepfakes which are largely similar to real media since deepfake generation techniques are still developing and improving rapidly. Furthermore, the arms race between the much-advanced team of deepfake creators and technology detectors gives an insight into the growing of the need towards the forensic techniques.

3. Methods And Materials

3.1 Proposed Methodology

The strategy to deal with deepfake pictures need to be whole enough that covers such activities as data collection and preprocessing, model choice, configuration, training, evaluation, computation efficiency assessment, interpretability analysis and comparisons.

In the first step, a mixed dataset consisting of both genuine and fake and a wide spectrum of manipulation types need to be prepared. The set of labeled data, augmented with transformations such as rotation, scaling and flipping, overcomes model overfitting because it improves the model training process by increasing the data diversity. Moreover, the details of labeling are useful to discriminate between the real and the crude deepfakes, hence, basis for the supervised learning.

Model Attention selection phase, the main deep learning architectures, such as EfficientNet, Inception-ResNet, and MobileNet, are reasonably chosen based on their innovative features, computational capabilities, and

execution suitability for image recognition. The use of the initial pre-trained weight renders transfer learning possible as prior knowledge is henceforth utilized. Fine-tuning of the model on the deepfake dataset, thus, is the training towards the specific deepfake identification and hyperparameter selection which is the experimental optimization of the performance through the iterative process.

Measurable evaluation metrics like accuracy and as an additional performance metrics we have AUC-ROC provide the models' performance with confusion matrices displays that aid in understanding the classification results. Computational efficiency analyze represents the comparison of selected inference speeds which will be then used to assess the usability of those architectures for speed cases with the accuracy/speed trade-off in mind.

Interpretability analysis occupies the central place of the analysis of evidences within forensic examination, which include searching through saliency maps and activations of features to interpret the decisions of model and identify the regions of interest in the images. This improves model independently explainability and quality, which help deepfake detection as well as understand manipulation.

Last but not the least EfficientNet- InceptionResNet- MobileNet analysis of performance will show their strengths, weaknesses, and the parts they are sacrifices in deep fake recognition. In this regard, the obtained insights are viewed as contributing to the improvement of the effectiveness of deepfake detection technologies and reinforcing defense strategies toward sophisticated digital forgeries.

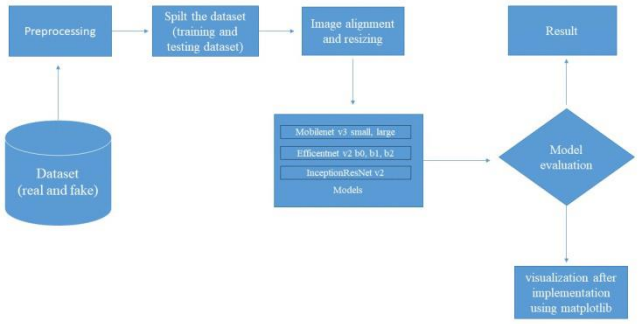

Figure 1. Proposed Architecture design

The workflow of a deepfake image identification system can be generally divided into multiple major stages as illustrated in (Figure 1). The process starts typically from inputting the image or a set of images subjected to the analysis. These pictures, after that, can be taken by different deep learning models, like EfficientNet, InceptionResNet, or MobileNet and they can serve as the backbone of the system by acting as feature extractors. Generally speaking, the role of this module is to find out and pick out the features which may be signs of deepfake alteration in the input images.

Then, the classification head is built on top of the feature extractor by using a predictive architect which is to keep track about whether the image is a deepfake or not. This decision-making unit is the part of the system that is classified as the knock-out head and classified as using the extracted features to make it possible to know the risk of the input image being a deepfake.

During the training phase, the model gets trained by a dataset having both authentic and deepfakes images, which have been tagged. This process is made up of a feature extractor and a classification head parameters optimisation process being carried out through the presentation of labelled data in a repetitive form that aims at actively teaching the model to correctly classify between real and manipulated images.

Finally, after model training, the output is the model score that determines the level of confidence of image authenticity. The issued likelihood score
can be considered a result of the forecasting system as a whole, in turn allowing the system to determine the data integrity alongside the degree of confidence in the output that there has been either no alteration or the alterations are present in the image. In sum, this workflow follows the iterative process undertaken by the deepfake image detection model to scrutinize the input images and eventually provide reliable predictions of the authenticity of the images.

3.2. Deepfake Dataset

The increasing accuracy of deep-fake applications, which are driven by neural networks and advanced imaging algorithms, has made it easier to craft highly realistic fake features. Deepfakes mergers such as those that use generative adversarial networks (GANs) to caption one person's face onto another's body, present to society a list of serious challenges including a breach of privacy, security, and a failure in the credibility of visual media. Accordingly, the academicians and expert practitioners have multiplied their efforts in designing dependable and deepfake detecting classifiers.

The datasets are essential in training deepfake recognition algorithms and in the evaluation process. Datasets of the authentic videos along with the deepfake videos, allowing us to label machine learning models for training and testing as ground truth. While having target assets of high quality, which contain various content and realistic imitation samples is great, however, such kinds of data are hard to obtain.

Kaggle - a renowned competition and cooperation platform for data professionals - has a wide range of data datasets across diverse domains. Kaggle has grown to be a highly-valued resource for researchers in the quest of deepfake detection and the related tasks in the recent years. On Kaggle one will find datasets including professionally prepared curated collections of deep fake videos that will be useful to investigators in training their models and conducting benchmark evaluations.

Figure 2. Sample dataset for deepfake detection taken from kaggle

The size, diversity and quality of Kaggle deep fake datasets are considered, which are key features of such data. Figure 2 shows the part of dataset taken from kaggle (Courtesy: https://www.kaggle.com/datasets/xhlulu/ real- and fake-faces-140k). Furthermore, we looked at the details of the metadata that came with each dataset like video resolution, frame rate and the annotation details. Having the ability to identify these features is critically important for the researchers to select a dataset precisely and preprocess it well.

3.3 Layers Used In Proposed System

The advent of neural networks implies a revolution in data-driven decision-making, fostering the growth of computer vision, natural language processing as well as of many other. Core to these networks ' performance are base layers that solve specific operations which, in effect, shape the network's ability to extract features of meaning

from data. The desire is to delve deeper into these key layers, unravelling their functions ad impacts on network training, design as well as includes algorithms elaborated in section 3.4.

Dense Layer: The Dense Layer provides an underpinning for neural networks establishing connections between units in adjacent layers. The dense connectivity of this layer is what allows it get information across; it also ads up and the network to learn the intricate relationships within the data. We investigate the dense layers performance by looking into such topics as the number of neurons and feedback function, and how they affect the network performance.

Activation Layer: Activation functions are the key to the non-linear mode for neural networks that let the systems model more complicated relationships and learn diverse patterns. In this subtopic, we consider activation functions such as ReLU, sigmoid, tanh and softmax. Discussion on their properties follows as it relates to the suitability to different tasks. Lastly, we also discuss the Activation layer which function is to apply those transformations to the output of preceding layers stating the Activation layer has an important role in the training of the network.

Dropout Layer: Overfitting is one of the most important problems in training neural networks and is essentially the reason why neural networks don't generalise well. On the other hand, the Dropout layer does one particular job which is randomly shut down a fraction of the neurons during the training process, therefore, the network won't adopt the specific patterns. We study ways Dropout regularization works, what it does to network trustworthiness, and how one can use it to good effect.

Conv2D Layer: Convolutional neural networks (CNNs) have become an indispensable component of computer vision and image recognition tasks with their unique Conv2D layer's design. These layers operate 2D convolution on input data, thus getting that spatial features of utmost importance for seeing visual information. We get in the concepts of convolutional functions, talking about the contribution of a convolutional filter in feature extraction and the effect of spatial hierarchies in CNN structures.

MaxPooling2D Layer: Max-pooling acts as an important component in the reduction of computational burden and the emission of core features in convolutional neural networks. Through the process of reduced-dimensional modeling of input, but holding onto significant information, MaxPooling2D layers provides the networks with translation-invariant characteristics, which play crucial role in the robustness of feature extraction. We talk about max pooling mechanisms along with the efficiency impact on the network and performance.

BatchNormalization Layer: The BatchNormalization layer is the possibility to deal with difficulties arising in deep neural networks by normalizing the activation of each batch. By minimizing the covariate shifts within, this level helps facilitate the convergence, improve the stability and enable generalization of the network modeling. We delve into the functionalities of batch normalization and its contribution to neural network structures; hence, its role in the training of neural networks as well as performance is emphasized.

3.4 Algorithms Used In The Proposed Solution

3.4.1 MobileNet V3 Small

Step 1: Input Processing

Input: An input image I of size WxHxC

Normalize: Firstly, subtract mean and then divide by standard deviation. Overall, this step enables normalization of the input image by subtracting the mean value and dividing by standard deviation such that input values come on a similar scale which, in turn, reduces the complexity of training.

Step 2: Feature Extraction

Convolution: Do a 3x3 convolution on the image with stride 2. As a result, a convolutional layer applies a 3x3 filter to an input image with a stride of 2, which leads to the creation of a feature map with reduced dimensions.

Inverted Residual Blocks:

For each block:

Pointwise Convolution: Complete conv2d(1, 1, 'same') with ReLU activation. This layer at the same time use the convolution with 1x1 kernel and ReLU activation function to reduce the number of channels (C).
Depthwise Separable Convolution:
Depthwise Convolution: Use a 3x3 depthwise separable convolution with ReLU activation. In Depthwiseconvolution, a distinct convolutional operation is applied to every channel within the input feature map, followed by a pointwise convolution to combine the results.
Pointwise Convolution: Follow with a 1x1 linear pointwise convolution. This step further reduces the number of channels through another 1x1 convolution without any non-linear activation function.
Step 3: Global Average Pooling
Pooling: Apply global average pooling to reduce spatial dimensions. Global average pooling calculates the average value for each feature map, yielding a singular value for each channel, effectively reducing the spatial dimensions to 1x1.
Step 4: Classifier
Fully Connected Layer:Add a fully connected layer. This layer establishes connections between every neuron in the preceding layer and every neuron in the following layer, facilitating the model's ability to comprehend intricate patterns.
Softmax Activation: Apply softmax activation for classification. The softmax activation function transforms the raw output scores from the preceding layer into probabilities, making it suitable for multi-class classification tasks.

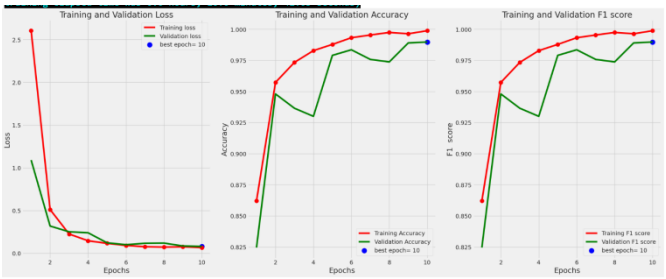

Figure 3. Training and validation loss, accuracy and F1 score for Mobilenet V3 Small algorithm

Figure 3 depicts is a line chart depicting training and validation loss, accuracy, and F1 score for the Mobilenet V3 Small algorithm. The chart includes information on best epoch, training loss, validation loss, accuracy, and F1 score over different epochs which concludes that the accuracy is 98.95%.

3.4.2 MobileNet V3 Large

Step 1: Input Processing
Input: An input image I of size WxHxC
Normalize: By subtracting the mean and dividing by the standard deviation, this process standardizes the input image. It ensures that the input values share a consistent scale, ultimately enhancing the training procedure.

Step 2: Feature Extraction
Convolution: Perform a larger convolution (e.g., 5x5) with increased stride (e.g., 2) for initial feature extraction. This larger convolutional layer applies a 5x5 filter to the input image with a larger stride of 2, resulting in a downsampled feature map.
Inverted Residual Blocks:
For each block:
Pointwise Convolution: Apply a larger 1x1 pointwise convolution with ReLU activation. This layer reduces the number of channels (C) by using a larger 1x1 convolutional layer is succeeded by a ReLU activation function.
Depthwise Separable Convolution:

Depthwise Convolution: Use a larger 3x3 depthwise separable convolution with ReLU activation. Depthwise convolution involves applying an individual convolutional operation to each channel within the input feature map, followed by a pointwise convolution to combine the results.
Pointwise Convolution: Follow with a larger 1x1 linear pointwise convolution. This step further reduces the number of channels through another 1x1 convolution without any non-linear activation function.

Step 3: Global Average Pooling
Pooling: Apply global average pooling to reduce spatial dimensions. Global average pooling calculates the average value for each feature map, leading to a solitary value per channel, effectively reducing the spatial dimensions to 1x1.

Step 4: Classifier
Fully Connected Layer: Add a complete connected layer with increased units. This layer establishes connections between each neuron in the preceding layer and every neuron in the subsequent layer, empowering the model to grasp intricate patterns with a larger number of parameters.
Softmax Activation: Apply softmax activation for classification. Softmaxfunction for activation converts the raw output scores of the preceeding layer into probabilities, making it suitable for multi-class classification tasks

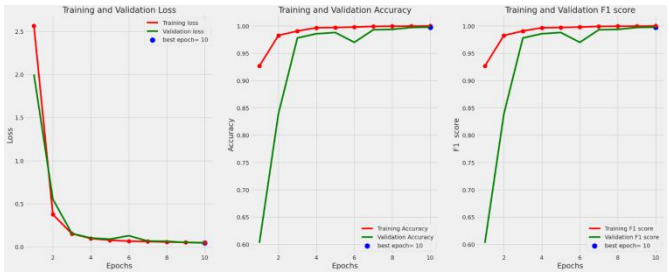

Figure 4. Training and validation loss, accuracy and F1 score for Mobilenet V3 Large algorithm

Figure 4 shows the graphs depicting training and validation loss, accuracy, and F1 score for the Mobilenet V3 Large algorithm. From the graph it is inferred that the accuracy is measured as 99.81%.

3.4.3 InceptionResNetV2

Step 1: Input Processing
Input: An input image I of size WxHxC
Normalize: Normalize the input image by subtracting its mean and dividing by its standard deviation. This procedure standardizes the input values, ensuring a consistent scale and potentially enhancing the training process.

Step 2: Stem Block
Utilize a sequence of convolutional and pooling layers to extract fundamental features and decrease spatial dimensions.
Convolutional layers: These layers utilize a collection of trainable filters to extract features from the input image.
Pooling layers: These layers shrink the spatial dimensions of the feature maps by downsampling, typically using max pooling or average pooling operations.

Step 3: Inception Modules
Utilize multiple Inception modules, which are parallel threads that each employ distinct convolutional operators(1x1, 3x3, 5x5 convolutions and pooling).
1x1 Convolution: Thus a 1x1 filter is applied to the feature maps, so that non-linear dimensionality reduction transformation is received.

3x3 Convolution: Such a convolutional layer attaches a 3x3 filter to the feature maps, detects the spatial aspect at an intermediate scale.
5x5 Convolution: This filter selection size is 5x5 and goes over the input feature maps, getting more information which is at a bigger scale.
Pooling: One of the primary functions of pooling operations which is either max pooling or average pooling is to reduce the size of input feature maps.
Concatenate the outputs of each branchs to form complex representations of the features.

Step 4: Residual Connections
Add residual connections into the Inception modules to enable the passage of gradients and accelerate the processes of training deep networks.Residual connections involve combining the input of a layer with its output, helping to mitigate the vanishing gradient problem and allowing for easier training of deeper networks.
Step 5: Reduction Blocks
Use convolutional layers with stride 2 and pooling operations to reduce spatial dimensions and increase feature depth. Reduction blocks typically involve applying convolutional layers with larger strides or pooling operations To decrease the spatial dimensions of the feature maps and simultaneously enhance the number of feature channels through downsampling.

Step 6: Global Average Pooling
Utilize global average pooling to amalgamate spatial details and condense the feature map into a singular vector. This process entails computing the average value for each feature map, ultimately condensing the spatial dimensions to 1x1.

Step 7: Classifier
Introduce a densely connected layer with softmax activation for the purpose of classification. It creates connections between every neuron in the preceding layer and all of the neurons in the following one and thus enables the model to comprehend complicated patterns, followed by softmax activation to output class probabilities for classification.

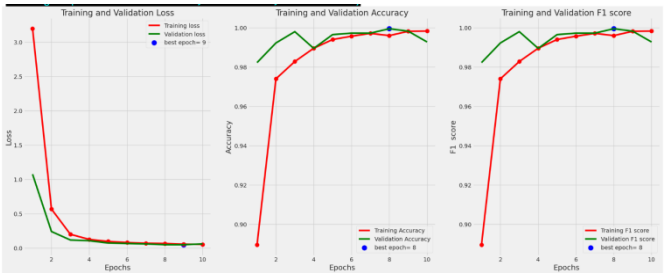

Figure 5. Training and validation loss, accuracy and F1 score for InceptionResNetV2 algorithm

Figure 5 depicts is a line chart depicting training and validation loss, accuracy, and F1 score for the InceptionResNetV2 algorithm. The chart includes information on best epoch, training loss, validation loss, accuracy, and F1 score over different epochs which conclude that the accuracy is 99.17%.

3.4.4 EfficientNetV2 B0

Step 1: Input Processing
Input: An input image I of size WxHxC
Normalize: Remove the population mean and divide by the standard deviation. In this step, the mean of the input image is subtracted and divided by the standard deviation to make sure that for training the data have similar scale, which allows the training process to run smoothly.
Step 2: Stem Block

Do conditional convolutions on the basis of feature extraction and reduce the spatial dimensions by layers. Convolutional layers: This concept refers to applying learnable filters to an image to sort out the peculiarities. Activation functions: In a neural network, ReLU(Rectified Linear Unit) activation functions would be introduced for non-linearity after every convolutional layers.

Step 3: Inverted Residual Blocks
For each block:
Depthwise Separable Convolution:
Depthwise Convolution: This is the next layer that divides the convolutional operation to the separated input channel for each feature map.
Pointwise Convolution: Finally we will pass through a 1x1 convolution for the grouping of them across channels.
Squeeze-and-Excitation Block:
Squeeze operation: This step involves global average pooling to aggregate spatial information across feature maps.
Excitation operation: Followed by fully connected layers with non-linear activations (e.g., ReLU) to model interdependencies between channels. This recalibrates the feature responses across channels to emphasize important features.

Step 4: Global Average Pooling
Use global average pooling to aggregate geometry knowledge by vectorizing the feature map. In a global average pooling, there are two different computations happening at the same time which makes a single value for each channel simply by reducing the spatial dimensions to 1x1.

Step 5: Classifier
An high fully connected layer with softmax activation can be adjusted to classification. Here, this densely connected layer is utilized to provide every neuron connected in the previous layer to every neuron in the forthcoming layer, thus making the model learn complex operations, and at last, softmax activation is used to have class probability prediction.

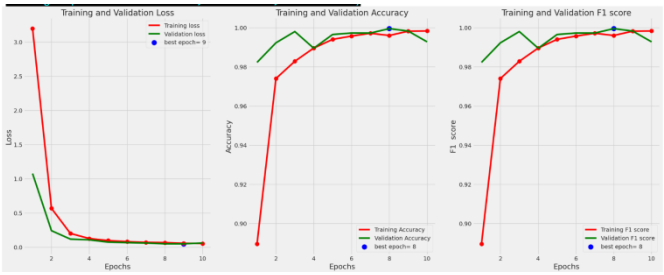

Figure 6. Training and validation loss, accuracy and F1 score for EfficientNetV2 B0 algorithm

Figure 6 depicts the graphs depicting validation and training loss, accuracy, and F1 score for the EfficientNetV2B0 algorithm. From the graph it is inferred that the accuracy is measured as 99.75%.

3.4.5 EfficientNetV2 B1 & B2
Step 1: Input Processing
Input: An input image I of size WxHxC
Normalize: Subtract the mean and divide by the standard deviation. This step standardizes the input image by subtracting the mean value and dividing by the standard deviation to ensure that the input values have a similar scale, which can improve the training process.

Step 2: Stem Block
Apply a series of convolutional layers to extract basic features and reduce spatial dimensions.
Convolutional layers: These layers apply learnable filters to the input image to extract features.

Activation functions: Typically after convolutional layers, the ReLU (Rectified Linear Unit) activation functions are applied to introduce non-linearity.

Step 3: Inverted Bottleneck Blocks
For each block:
Inverted Bottleneck Block:
Depthwise Separable Convolution:
Depthwise Convolution: In this layer for each channel of the input feature map a separate convolutional operation is applied.
Pointwise Convolution: Followed by a 1x1 convolution to combine the results across channels. This step helps to reduce computational cost and parameter size.
Squeeze-and-Excitation Block:
Across the feature maps to aggregate the spatial information Squeeze global average pooling is applied.

Excitation operation: Fully connected layers with non-linear activations (e.g., ReLU) are used to model interdependencies between channels. This recalibrates the feature responses across channels to emphasize important features.

Step 4: Global Average Pooling
To aggregate spatial information and reduce the feature map to a single vector apply global average pooling. Global average pooling computes each feature map by average value, resulting in single value for each channel and this leading to 1x1 spatial dimensions.
Step 5: Classifier
Add a fully connected layer for classification with softmax activation. This fully connected layer connects every neuron of previous layer to every neuron in the subsequent layer, aids the model to understand the complex patterns, followed by softmax activation to output class probabilities for classification.

Figure 7 and Figure 8 depicts is a line chart depicting training and validation loss, accuracy, and F1 score for the EfficientNetV2 B1 & B2 algorithm. The chart includes information on best epoch, training loss, validation loss, accuracy, and F1 score over different epochs which conclude that the accuracy is 99.83% and 99.80% respectively.

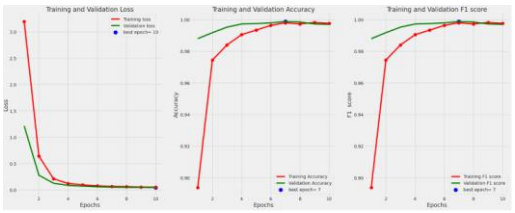

Figure 7. Training and validation loss, accuracy and F1 score for EfficientNetV2 B1 algorithm

Figure 8. Training and validation loss, accuracy and F1 score for EfficientNetV2 B2 algorithm

4. Result And& Discussions

In this study, the evaluation and analysis of a deepfake detection technique are presented, aiming to address the growing concerns surrounding the proliferation of manipulated media. The effectiveness of the proposed

technique is assessedusing EfficientNetV2B1 algorithm through rigorous testing, as illustrated in Figures 9 through 12.

Epoch	Train Loss	Train Accuracy	Valid Loss	Valid Accuracy	V_Loss % Improvement	Learning Rate	Next LR Rate	Duration in Seconds
1	3.1931	89.38	1.2136	98.80	0.00	0.001000	0.001000	176.58
2	0.6430	97.44	0.2786	99.18	77.04	0.001000	0.001000	151.25
3	0.2115	98.40	0.1273	99.52	54.32	0.001000	0.001000	150.57
4	0.1218	99.05	0.0846	99.73	33.53	0.001000	0.001000	150.90
5	0.0956	99.34	0.0716	99.75	15.41	0.001000	0.001000	150.74
6	0.0762	99.64	0.0596	99.80	16.76	0.001000	0.001000	150.96
7	0.0663	99.80	0.0517	99.87	13.22	0.001000	0.001000	151.21
8	0.0621	99.73	0.0474	99.85	8.36	0.001000	0.001000	151.29
9	0.0542	99.83	0.0472	99.73	0.30	0.001000	0.001000	151.52
10	0.0529	99.76	0.0449	99.70	4.93	0.001000	0.001000	151.66

Figure 9. Improvement of test class on basis of training class mentioning its accuracy and loss

Figure 9 showcases the improvement of the test class concerning the training class, emphasizing accuracy and loss metrics. The results reveal a significant advancement, indicating the efficacy of the employed methodology. Notably, the accuracy attained in this evaluation process is an impressive 99.83%, underscoring the reliability and robustness of the deepfake detection model.

In Figure 10, a comprehensive classification report is provided, detailing the performance of the proposed technique across 10,000 samples. This report offers valuable insights into various evaluation metrics such as precision, recall, and f1-score, further corroborating the high accuracy achieved by the model.

Classification Report:

	precision	recall	f1-score	support
fake	0.9998	0.9968	0.9983	10000
real	0.9968	0.9998	0.9983	10000
accuracy			0.9983	20000
macro avg	0.9983	0.9983	0.9983	20000
weighted avg	0.9983	0.9983	0.9983	20000

Figure10. Classification report of the proposed deepfake detection technique of 10000 samples

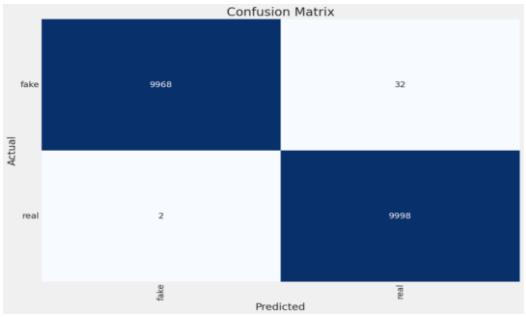

Figure 11. Confusion matrix

Furthermore, in Figure 11 the confusion matrix can be seen. It represents the performance of the deepfake detection system. The confusion matrix is explained into 4 variables as true positives, true negatives, false positives, and false negatives. The Figure 11 offers a clear output and clarity of the models abilityin finding the difference between altered and unaltered media.

Lastly, Figure 12shows the bar graph on the classification errors on the test dataset, highlighting the number of misclassifications for real and fake. By examining the errors from the graph, researchers gain valuable knowledge and research even more for advancement in the deepfake detection framework.

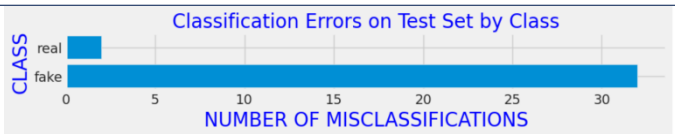

Figure12. Classification Errors on test dataset based on the number of miscalculation

Thevarious algorithms described in the proposed work aids in the importance of using the most effective model for the given task. The table (Table 1)shows the comparison between different algorithms used based on the metricslike Precision, Recall, F1 Score, and Accuracy. This valuable insightgives a clear clarity on the performance of different algorithms. Notably, EfficientNetV2B1 emerges as the most effective model having highest score. This highlights the superior capability of EfficientNetV2B1in comparison with other algorithms and metrics i.e.,MobileNet V3 (both small and large variants), EfficientNetV2B0, EfficientNetV2B2, and InceptionResNetV2.

EfficientNetV2B1's higher performance can be due to its advanced architecture, which gives a balance between model complexity and computational efficiency. EfficientV2B1 gives an exceptional output in the metrics i.e., precision, recall, f1-score and accuracy by using the compound scaling and neural architecture search techniques.Even models such as MobileNet V3 and other EfficientNetV2 variants, also demonstrate strong performance but compared to EfficientV2B1 they have slight difference in performance.

Table 1. Comparison of metrics of different algorithmsused in the proposed work

Algorithms	Precision	Recall	F1 Score	Accuracy
MobileNet V3 – Small	98.95	98.95	98.95	98.95
MobileNet V3 – Large	99.81	99.80	99.80	99.81
EfficientNetV2B0	99.76	99.76	99.75	99.75
EfficientNetV2B1	99.83	99.83	99.83	99.83
EfficientNetV2B2	99.80	99.80	99.79	99.80
InceptionResNetV2	99.18	99.18	99.17	99.17

5. Conclusion

The other side of the coin was preparing as the disruption itself was just building up. As such, the faithful performance of EfficientNetV2, MobilenetV3, and InceptionResNetV2 was vital in giving a solution, and therefore, deep learning was the sensible choice. Which is being done by an extended network type of this EfficientNetV2B1 which is not natural is that it is one of the networks. Besides, the demonstrated model seems better in comparison to other models like F-Score, precision, recall of POS tags and accuracy that evaluate the models as mentioned earlier. It could be the basis for developing our researches. The neural network architecture as the EfficientNetV2B1- which is more curious and seeker about every specifics of a human being's facial image by nature also contributes to make a progress in the deep learning field via enabling the process to be executed effectively. Moreover, the approach used is a kind of algorithm which is designed directly to the deepfake problem by precise gratification of parameters and the relevant computational platform can boost the diversity of different kinds of distortion or fake image ways. Likewise, AI would be capable of identifying medical conditions, providing medical advice and even performing surgeries at a faster rate without the risk of human error. His personal experience, of having learned diverse metrics helps him hold with confidence that the AI models like himself, including the models which he would be creating till his death would go on to win the competition of improving the AI capability. EfficientNetV2-B1 will become the spotlight, hands down, it is exactly the substance, the anchor, which will give intellectually manipulated or fake output the life, it needs. In my analysis about cybersecurity and its relationship to media integrity, the core element that makes the puzzle solution to this

question become clear is the deepfake contents. This is because the lifelike facades presented in an alternative but fake reality portray it as a viable option for consumption.

References

1. Taeb, M., & Chi, H. (2022, March 4). Comparison of Deepfake Detection Techniques through Deep Learning. Journal of Cybersecurity and Privacy, 2(1), 89–106. https://doi.org/10.3390/jcp2010007
2. Corcoran, K., Ressler, J., & Zhu, Y. (2021). Countermeasure against Deepfake Using Steganography and Facial Detection. Journal of Computer and Communications, 09(09), 120–131. https://doi.org/10.4236/jcc.2021.99009
3. Elhassan, A., Al-Fawa'reh, M., Jafar, M. T., Ababneh, M., & Jafar, S. T. (2022, July). DFT-MF: Enhanced deepfake detection using mouth movement and transfer learning. SoftwareX, 19, 101115. https://doi.org/10.1016/j.softx.2022.101115
4. Singh, A., Saimbhi, A. S., Singh, N., & Mittal, M. (2020, June 23). DeepFake Video Detection: A Time-Distributed Approach. SN Computer Science, 1(4). https://doi.org/10.1007/s42979-020-00225-9
5. Lee, E. G., Lee, I., & Yoo, S. B. (2023, September 17). ClueCatcher: Catching Domain-Wise Independent Clues for Deepfake Detection. Mathematics, 11(18), 3952. https://doi.org/10.3390/math11183952
6. Su, W. J. (2022, January 1). Boosting Deepfake Detection Via Hard Sample Mining and Augmentation.
7. Robust DeepFake Detection: Using Decoy Mechanism for Resisting Adversarial and Face Manipulation Attacks. (2022, January 1).
8. A Deepfake Detection Algorithm With Improved Generalization Ability. (2021, January 1).
9. Rathgeb, C., Tolosana, R., Vera-Rodriguez, R., & Busch, C. (2022, January 31). Handbook of Digital Face Manipulation and Detection. Springer Nature.
10. Obaid, A. J., Abdul-Majeed, G. H., Burlea-Schiopoiu, A., & Aggarwal, P. (2023, January 3). Handbook of Research on Advanced Practical Approaches to Deepfake Detection and Applications. IGI Global.
11. Yu, P., Xia, Z., Fei, J., & Lu, Y. (2021, April 9). A Survey on Deepfake Video Detection. IET Biometrics, 10(6), 607–624. https://doi.org/10.1049/bme2.12031
12. Groh, M., & Ramon, M. (2022, December 5). Do Super Recognizers Excel at Deepfake Detection? Journal of Vision, 22(14), 3993. https://doi.org/10.1167/jov.22.14.3993
13. Raza, A., Munir, K., & Almutairi, M. (2022, September 29). A Novel Deep Learning Approach for Deepfake Image Detection. Applied Sciences, 12(19), 9820. https://doi.org/10.3390/app12199820
14. Xue, Z., Liu, Q., Shi, H., Zou, R., & Jiang, X. (2022, December 12). A Transformer-Based DeepFake-Detection Method for Facial Organs. Electronics, 11(24), 4143. https://doi.org/10.3390/electronics11244143
15. Zhao, L., Zhang, M., Ding, H., & Cui, X. (2021, December 17). MFF-Net: Deepfake Detection Network Based on Multi-Feature Fusion. Entropy, 23(12), 1692. https://doi.org/10.3390/e23121692
16. Ismail, A., Elpeltagy, M., Zaki, M., &ElDahshan, K. A. (2021, September 21). Deepfake video detection: YOLO-Face convolution recurrent approach. PeerJ Computer Science, 7, e730. https://doi.org/10.7717/peerj-cs.730
17. Zhang, T. (2022, January 8). Deepfake generation and detection, a survey. Multimedia Tools and Applications, 81(5), 6259–6276. https://doi.org/10.1007/s11042-021-11733-y
18. Khormali, A., & Yuan, J. S. (2022, March 14). DFDT: An End-to-End DeepFake Detection Framework Using Vision Transformer. Applied Sciences, 12(6), 2953. https://doi.org/10.3390/app12062953
19. Tran, V. N., Kwon, S. G., Lee, S. H., Le, H. S., & Kwon, K. R. (2023). Generalization of Forgery Detection With Meta Deepfake Detection Model. IEEE Access, 11, 535–546. https://doi.org/10.1109/access.2022.3232290 37
20. Kim, Y. S., Song, H. J., & Han, J. H. (2022, January 20). A Study on the Development of Deepfake-Based Deep Learning Algorithm for the Detection of Medical Data Manipulation. Webology, 19(1), 4396–4409. https://doi.org/10.14704/web/v19i1/web19289
21. Khormali, A., & Yuan, J. S. (2021, September 27). ADD: Attention-Based DeepFake Detection Approach. Big Data and Cognitive Computing, 5(4), 49. https://doi.org/10.3390/bdcc5040049
22. Alnaim, N. M., Almutairi, Z. M., Alsuwat, M. S., Alalawi, H. H., Alshobaili, A., &Alenezi, F. S. (2023). DFFMD: A Deepfake Face Mask Dataset for Infectious Disease Era With Deepfake Detection Algorithms. IEEE Access, 11, 16711–16722. https://doi.org/10.1109/access.2023.3246661

23. Sohaib, M., &Tehseen, S. (2023). Forgery detection of low quality deepfake videos. Neural Network World, 33(2), 85–99. https://doi.org/10.14311/nnw.2023.33.006
24. Taeb, M., & Chi, H. (2022, March 4). Comparison of Deepfake Detection Techniques through Deep Learning. Journal of Cybersecurity and Privacy, 2(1), 89–106. https://doi.org/10.3390/jcp2010007
25. Deepfake Detection Using Xception and LSTM. (2023, June 7). International Research Journal of Modernization in Engineering Technology and Science. https://doi.org/10.56726/irjmets41123
26. European researchers advance deepfake detection. (2022, April). Biometric Technology Today, 2022(4). https://doi.org/10.12968/s09694765(22)70051-3
27. Fosco, C., Josephs, E., Andonian, A., & Oliva, A. (2022, December 5). Deepfake Caricatures: Human-guided Motion Magnification Improves Deepfake Detection by Humans and Machines. Journal of Vision, 22(14), 4079. https://doi.org/10.1167/jov.22.14.4079
28. Wubet, W. M. (2020, April 30). The Deepfake Challenges and Deepfake Video Detection. International Journal of Innovative Technology and Exploring Engineering, 9(6), 789–796. https://doi.org/10.35940/ijitee.e2779.049620
29. KORKMAZ, A., & ALKAN, M. (2023, July 5). Deepfake Video Detection Using Deep Learning Algorithms. PoliteknikDergisi, 26(2), 855–862. https://doi.org/10.2339/politeknik.1063104
30. Padir, O., Tandel, R., Thakare, S., & Bide, P. (2022). Deepfake in an Image Using Deep Learning. SSRN Electronic Journal. https://doi.org/10.2139/ssrn.4294519